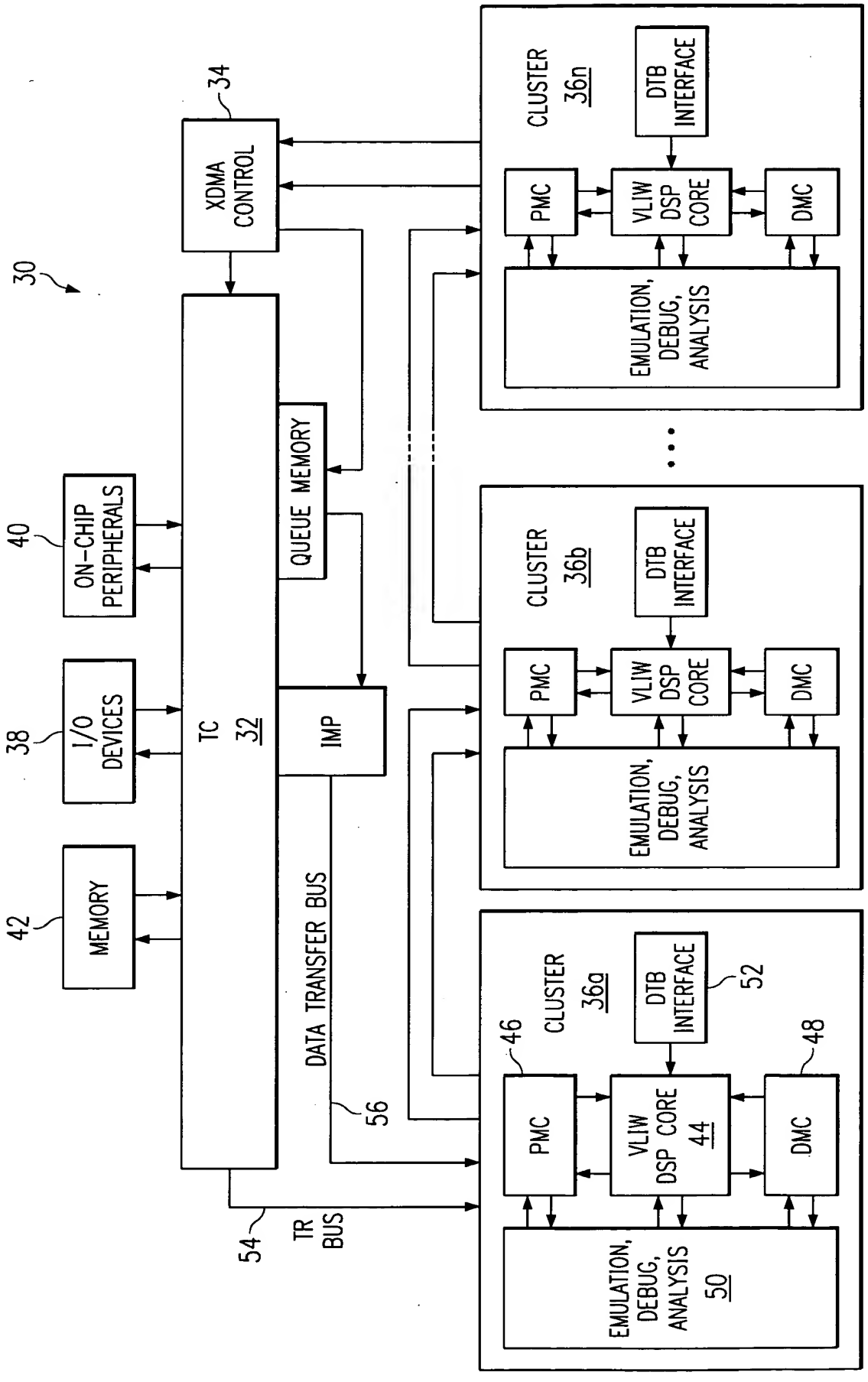
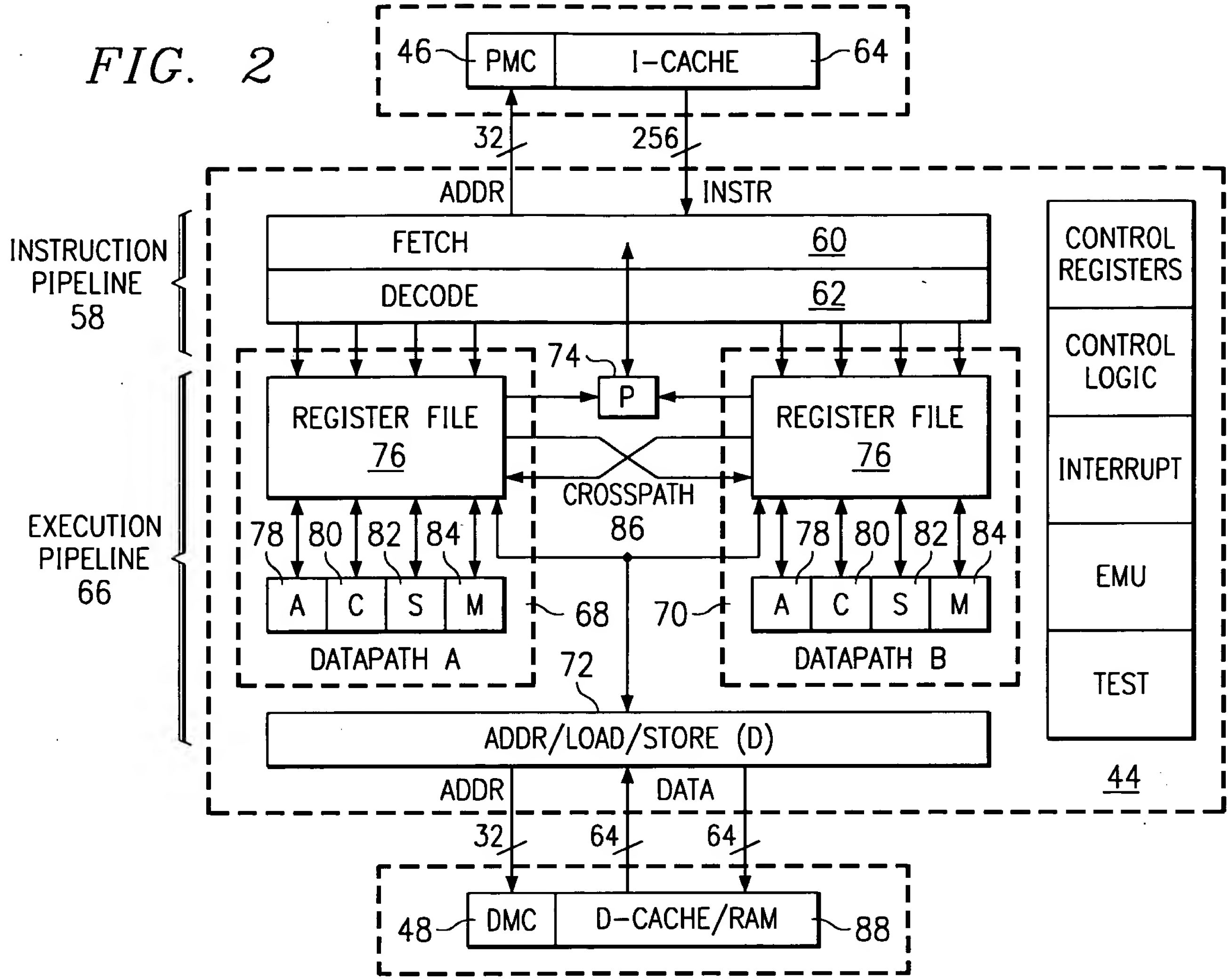

30
XDMA CONTROL
34
TC
32
QUEUE MEMORY
IMP
DATA TRANSFER BUS
56
TR BUS
54
ON-CHIP PERIPHERALS
40
I/O DEVICES
38
MEMORY
42
CLUSTER 36n
DTB INTERFACE
PMC
VLIW DSP CORE
DMC
EMULATION, DEBUG, ANALYSIS
CLUSTER 36b
CLUSTER 36a
46
48
52
VLIW DSP CORE 44
EMULATION, DEBUG, ANALYSIS 50

FIG. 1

FIG. 2



UNIT GROUP	OPERATIONS	REGISTER FILE ACCESS	
		PRIMARY DATAPATH	ALTERNATIVE DATAPATH
A	- GENERAL ARITHMETIC - BOOLEAN AND CONTROL REGISTER ACCESS	R/W	R
C	- COMPARE, SHIFT, BOOLEAN - ARITHMETIC: ADD, SUB	R/W	R
S	- SHIFT, ROTATE, EXTENDED BOOLEAN - ARITHMETIC: ADD, SUB	R/W	R
M	- MULTIPLY - ARTHMETIC: ADD, SUB	R/W	R
D	- LOAD - STORE - ADDRESS COMPUTATION	W TO BOTH R FROM BOTH R/W BOTH	
P	- BRANCH	R FROM BOTH	

FIG. 3

R=READ, W=WRITE

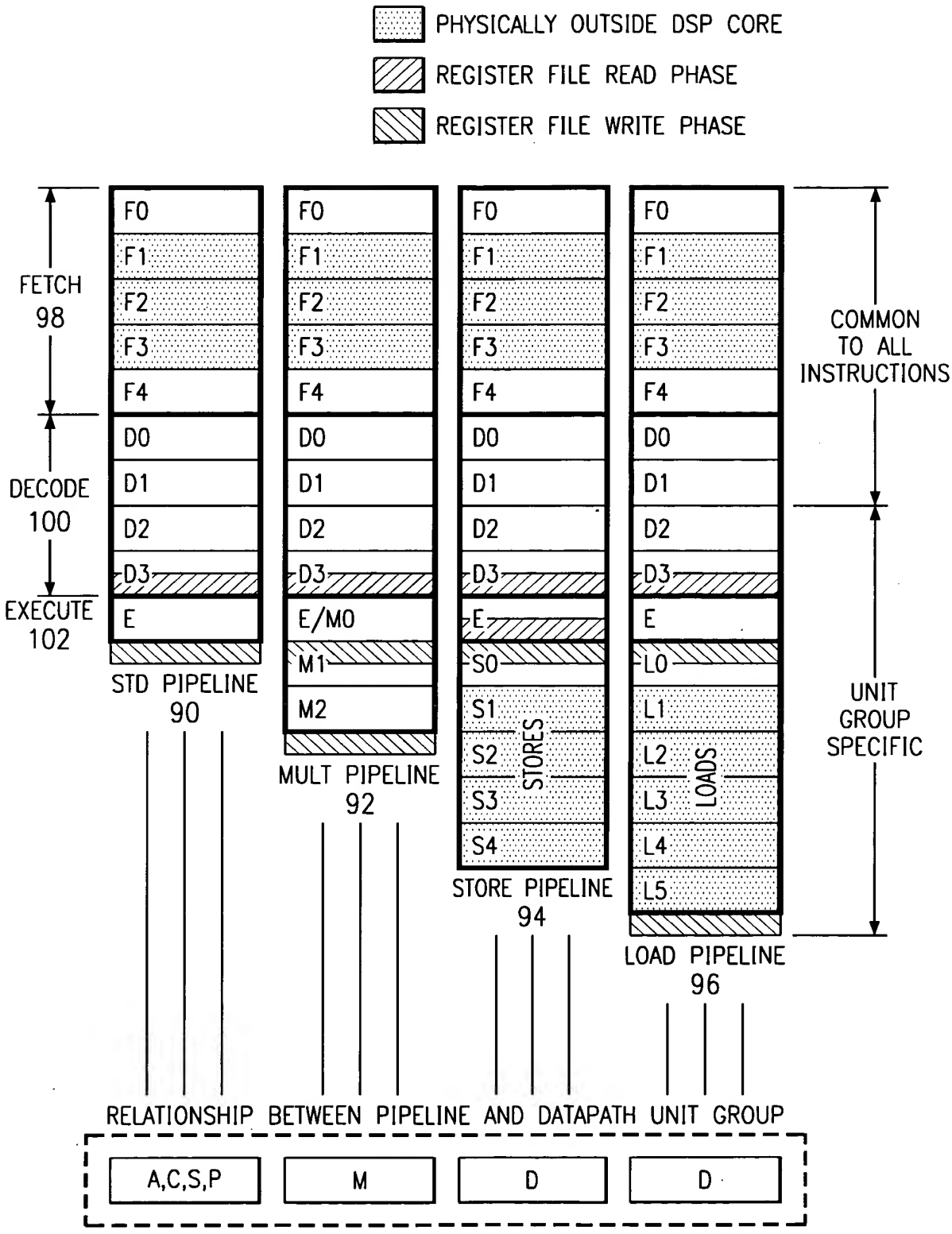

PHYSICALLY OUTSIDE DSP CORE
REGISTER FILE READ PHASE
REGISTER FILE WRITE PHASE
FETCH
98
DECODE
100
EXECUTE
102
F0
F1
F2
F3
F4
D0
D1
D2
D3
E
E/M0
M1
M2
S0
S1
S2
S3
S4
L0
L1
L2
L3
L4
L5
STORES
LOADS
COMMON TO ALL INSTRUCTIONS
UNIT GROUP SPECIFIC
STD PIPELINE
90
MULT PIPELINE
92
STORE PIPELINE
94
LOAD PIPELINE
96
RELATIONSHIP BETWEEN PIPELINE AND DATAPATH UNIT GROUP
A,C,S,P
M
D
D

FIG. 4

STAGE	FUNCTION
F0	SEND PC TO PROGRAM MEMORY CONTROLLER. LDIP ASSIGNED.
F1	CACHE BLOCK SELECT.
F2	ADDRESS PHASE OF INSTRUCTION CACHE ACCESS.
F3	DATA PHASE OF INSTRUCTION CACHE ACCESS.
F4	FETCH PACKET SENT TO DSP.

FIG. 5a

STAGE	FUNCTION
D0	DETERMINE VALID INSTRUCTIONS IN CURRENT FETCH PACKET.
D1	SORTS INSTRUCTIONS IN EXECUTE PACKET ACCORDING TO DESTINATION UNITS.
D2	INSTRUCTIONS SENT TO DESTINATION UNITS. CROSSPATH REGISTER READS OCCUR.
D3	UNITS DECODE INSTRUCTIONS. REGISTER FILE READ (2ND PHASE).

FIG. 5b

UNIT	STAGE	FUNCTION
NON M UNIT	E	EXECUTION OF OPERATION BEGINS AND COMPLETES. FULL RESULT AVAILABLE AT END OF CYCLE.
M UNIT	M0	EXECUTION OF MULTIPLY OPERATION BEGINS. (OR, NON-MULTIPLY OPERATION BEGINS AND COMPLETES.)
M UNIT	M1	MULTIPLY OPERATION CONTINUES. (OR, NON-MULTIPLY RESULT WRITTEN TO REGISTER FILE (PHASE 1).)
M UNIT	M2	MULTIPLY OPERATION COMPLETES.

FIG. 5c

STAGE	FUNCTION
E	ADDRESS GENERATION OCCURS. REGISTER FILE ACCESS FOR READ DATA.
L0	LOAD ADDRESS GENERATED DURING E IS SENT TOWARDS THE DMC.
L1	ADDRESS DECODE, TC ARBITRATION, TAG COMPARES.
L2	ADDRESS DECODE, TC ARBITRATION, TAG COMPARES.
L3	ADDRESS PHASE OF DATA CACHE ACCESS.
L4	DATA PHASE OF DATA CACHE ACCESS.
L5	64-BIT DATA SENT TO DSP.

FIG. 5d

STAGE	FUNCTION
E	ADDRESS GENERATION OCCURS. REGISTER FILE ACCESS FOR WRITE DATA.
S0	ADDRESS SENT TO DMC.
S1	ADDRESS DECODE IN DMC. WRITE DATA ALIGNMENT.
S2	TAG COMPARE IN DMC. WRITE DATA SENT TO DMC.
S3	ADDRESS PHASE IN DATA CACHE.
S4	DATA PHASE IN DATA CACHE.

FIG. 5e

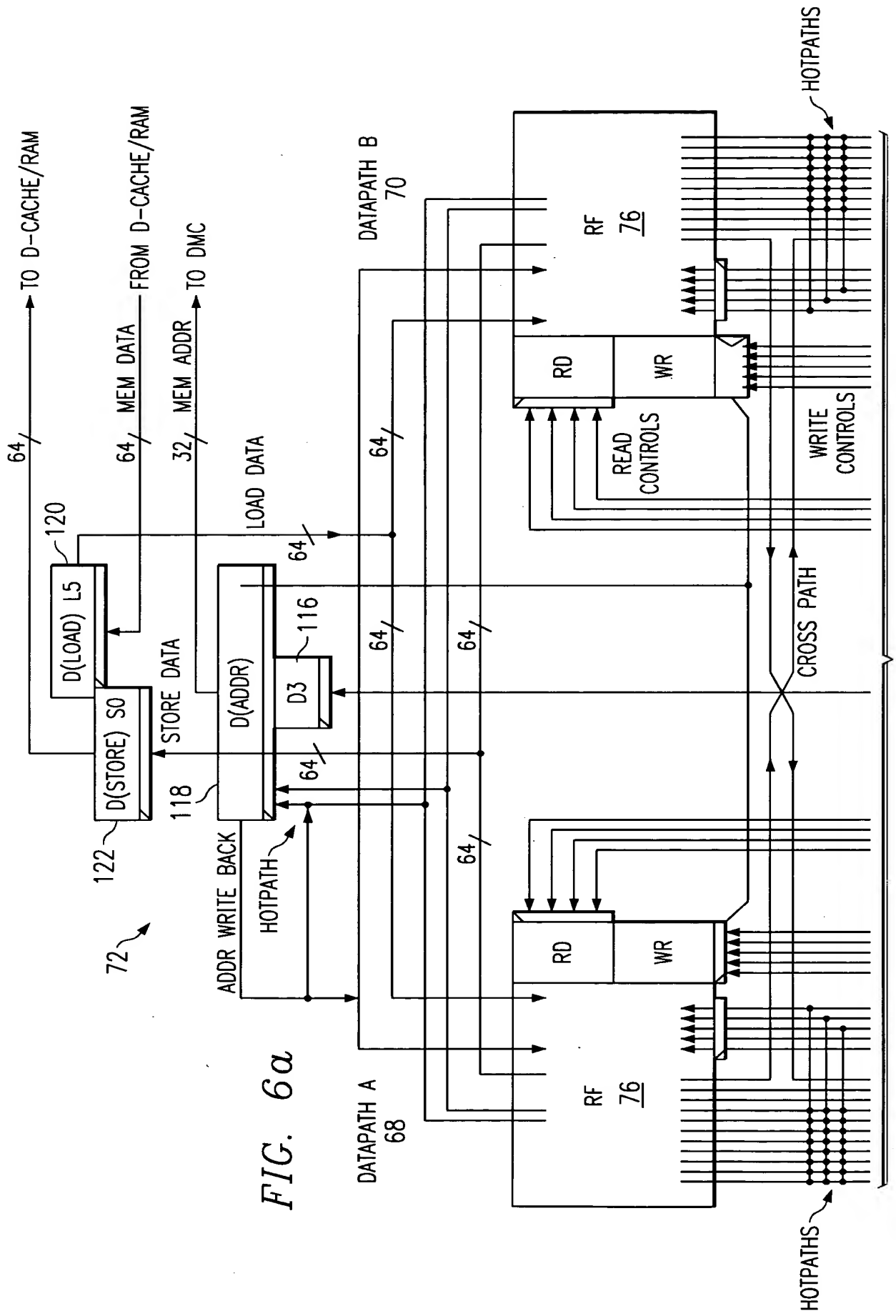

TO D-CACHE/RAM
FROM D-CACHE/RAM
TO DMC
64
64
MEM DATA
32
MEM ADDR
LOAD DATA
64
DATAPATH B
70
HOTPATHS
120
D(LOAD) L5
D(STORE) S0
STORE DATA
122
118
D(ADDR)
D3
116
64
64
64
64
64
RF
76
RD
WR
READ CONTROLS
WRITE CONTROLS
CROSS PATH
ADDR WRITE BACK
HOTPATH
72
DATAPATH A
68
RF
76
RD
WR
HOTPATHS
TO FIG. 6B

FIG. 6a

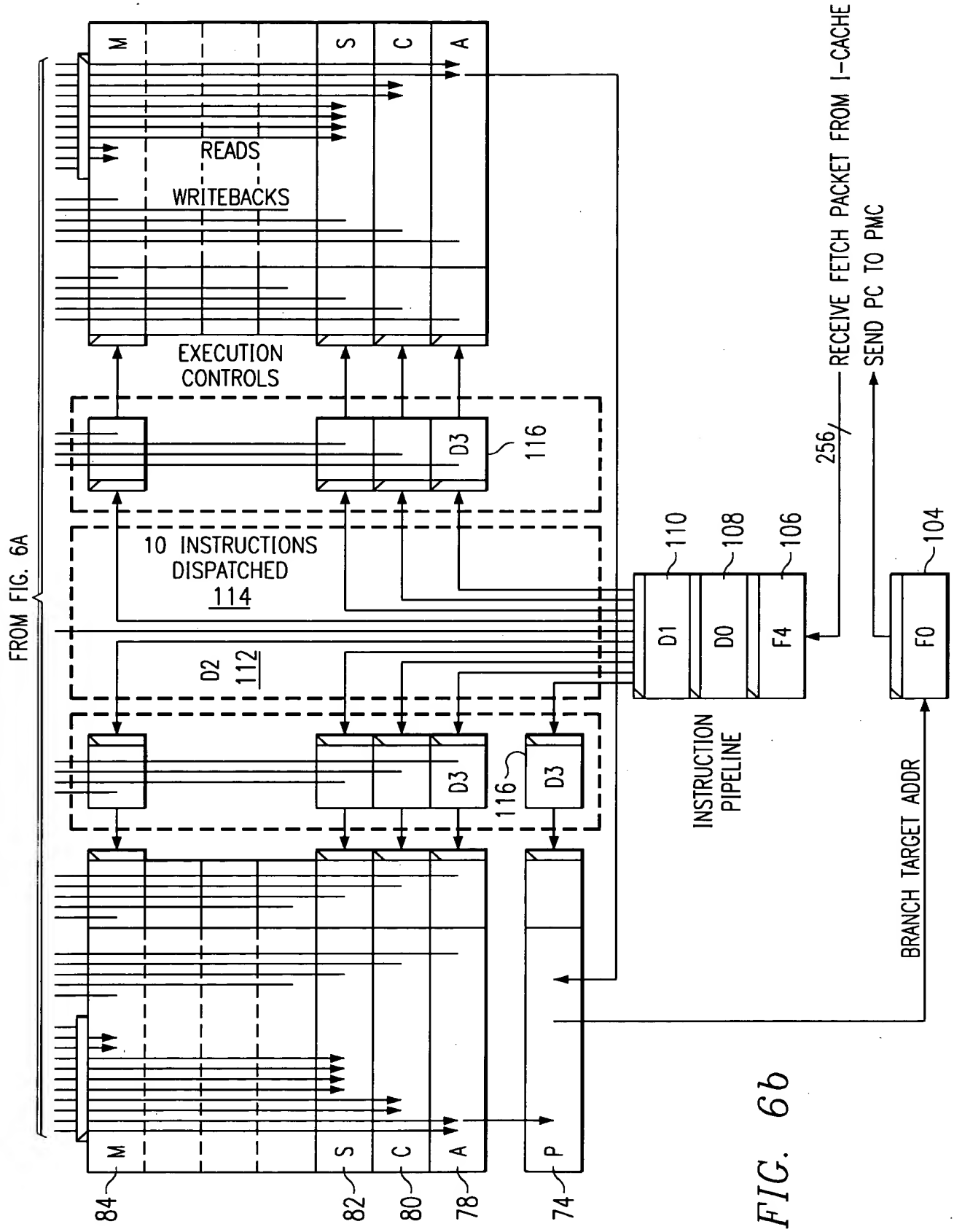

FROM FIG. 6A
M
S
C
A
READS
WRITEBACKS
EXECUTION
CONTROLS
D3
116
10 INSTRUCTIONS
DISPATCHED
114
D2
112
D1
D0
F4
F0
110
108
106
104
256
RECEIVE FETCH PACKET FROM I-CACHE
SEND PC TO PMC
INSTRUCTION
PIPELINE
BRANCH TARGET ADDR
P
84
82
80
78
74

FIG. 6b

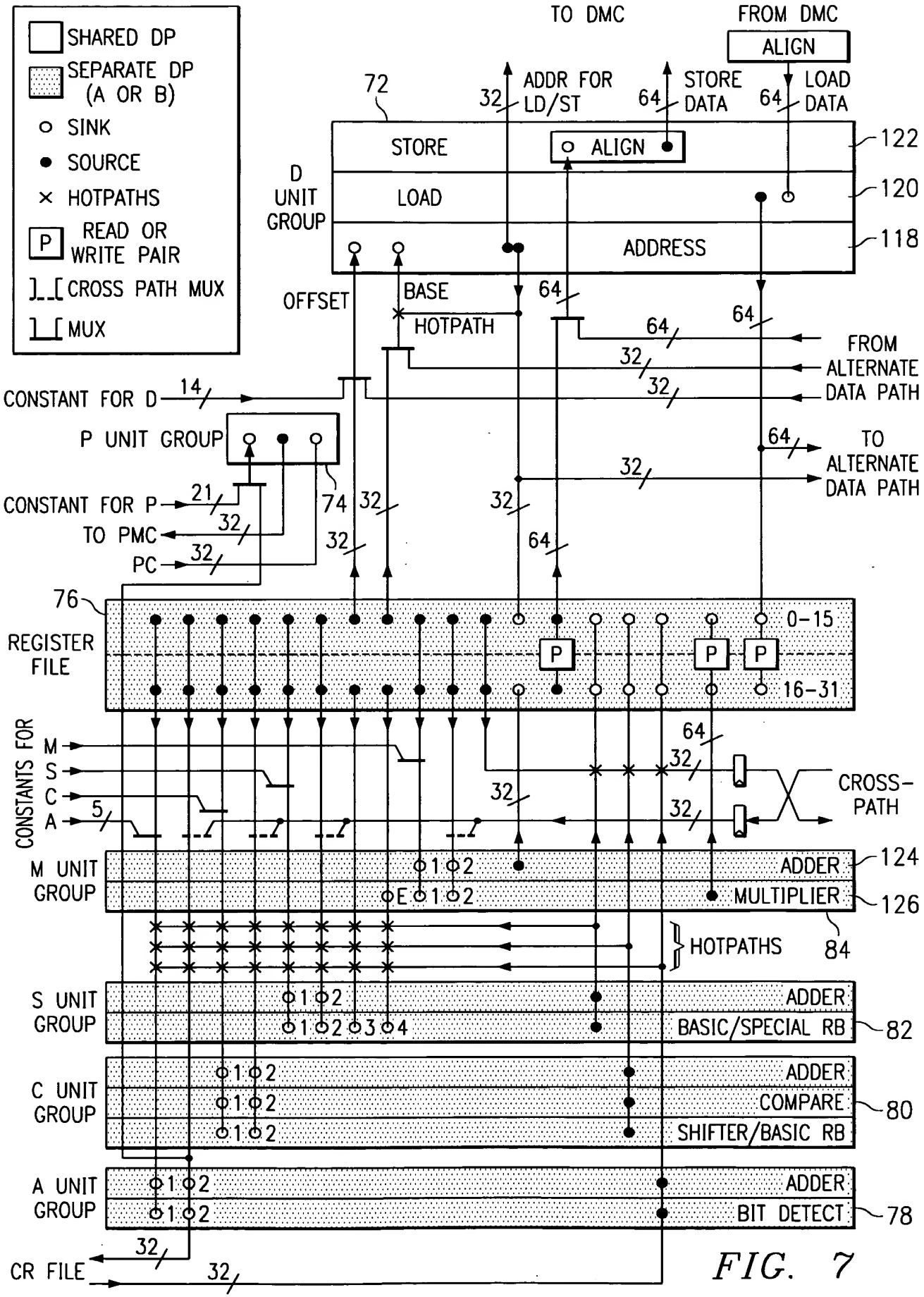

SHARED DP
SEPARATE DP (A OR B)
SINK
SOURCE
HOTPATHS
READ OR WRITE PAIR
CROSS PATH MUX
MUX
TO DMC
FROM DMC
ALIGN
72
32 ADDR FOR LD/ST
64 STORE DATA
64 LOAD DATA
D UNIT GROUP
STORE
ALIGN
LOAD
ADDRESS
122
120
118
OFFSET
BASE
HOTPATH
64
FROM ALTERNATE DATA PATH
CONSTANT FOR D
14
P UNIT GROUP
TO ALTERNATE DATA PATH
CONSTANT FOR P
21
74
TO PMC
PC
76
REGISTER FILE
0-15
16-31
CONSTANTS FOR
M
S
C
A
5
CROSS-PATH
M UNIT GROUP
ADDER
MULTIPLIER
124
126
HOTPATHS
84
S UNIT GROUP
BASIC/SPECIAL RB
82
C UNIT GROUP
COMPARE
SHIFTER/BASIC RB
80
A UNIT GROUP
BIT DETECT
78
CR FILE

FIG. 7

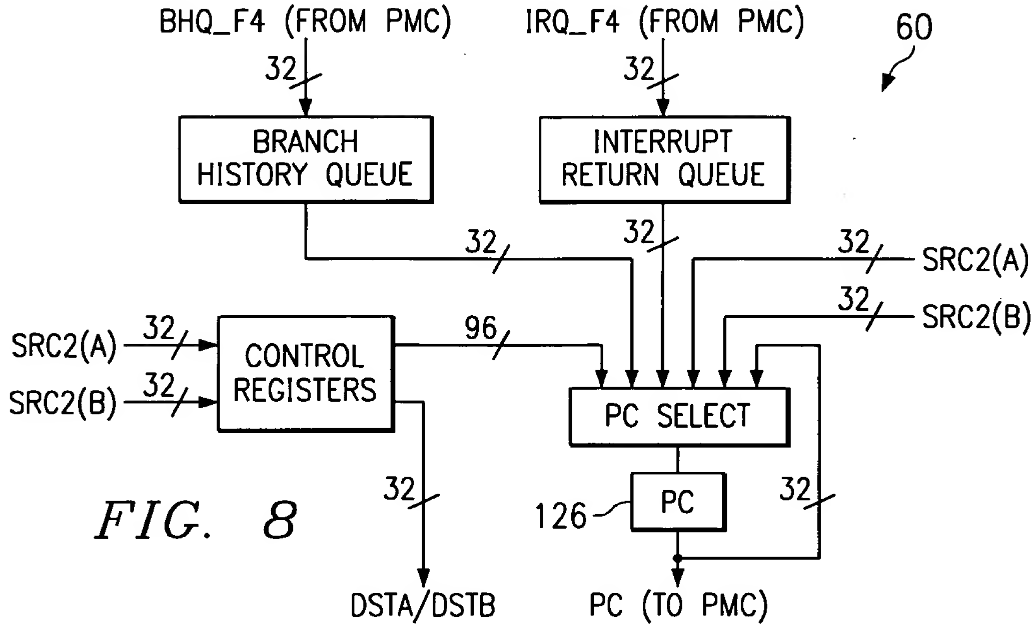


FIG. 8

|| [PREDICATION REG] INSTRUCTION_MNEMONIC .UNIT-DATAPATH-CROSSPATH OP1, OP2, DST

WHERE:

||	=TO BE SCHEDULED IN PARALLEL WITH PRECEDING INSTRUCTION(S)
[PREDICATION REG]	=REGISTER CONTAINING PREDICATION VALUE
.UNIT	=A,C,S,M,D,P UNIT GROUPS
DATAPATH	=1 FOR DATAPATH A, 2 FOR DATAPATH B
CROSSPATH	=X IF ONE OPERAND COMES FROM OPPOSITE REGISTER FILE
OP1, OP2	=SOURCE REGISTERS
DST	=DESTINATION REGISTER

UNIT GROUP	ASSEMBLY NOTATIONS		ASSEMBLY EXAMPLES	WITH CROSSPATH
	DATAPATH A	DATAPATH B		
A	.A1	.A2	ADD .A1 A1,A2,A3 SUB .A2 B1,B2,B3	ADD .A1X A1,B2,A3 SUB .A2X B1,A2,B3
C	.C1	.C2	CMPEQ .C1 A1,A2,A3 CMPEQ .C2 B1,B2,B3	CMPEQ .C1X A1,B2,A3 CMPEQ .C2X B1,A2,B3
S	.S1	.S2	SHL .S1 A1,A2,A3 SHL .S2 B1,B2,B3	SHL .S1X A1,B2,A3 SHL .S2X B1,A2,B3
M	.M1	.M2	MPY .M1 A1,A2,A3 MPY .M2 B1,B2,B3	MPY .M1X A1,B2,A3 MPY .M2X B1,A2,B3
D	.D		LDB .D *A8,A12 STB .D A8,*A12 ADDAH .D A8,A2,B1	n/a
P	.P		B A8	n/a

FIG. 15

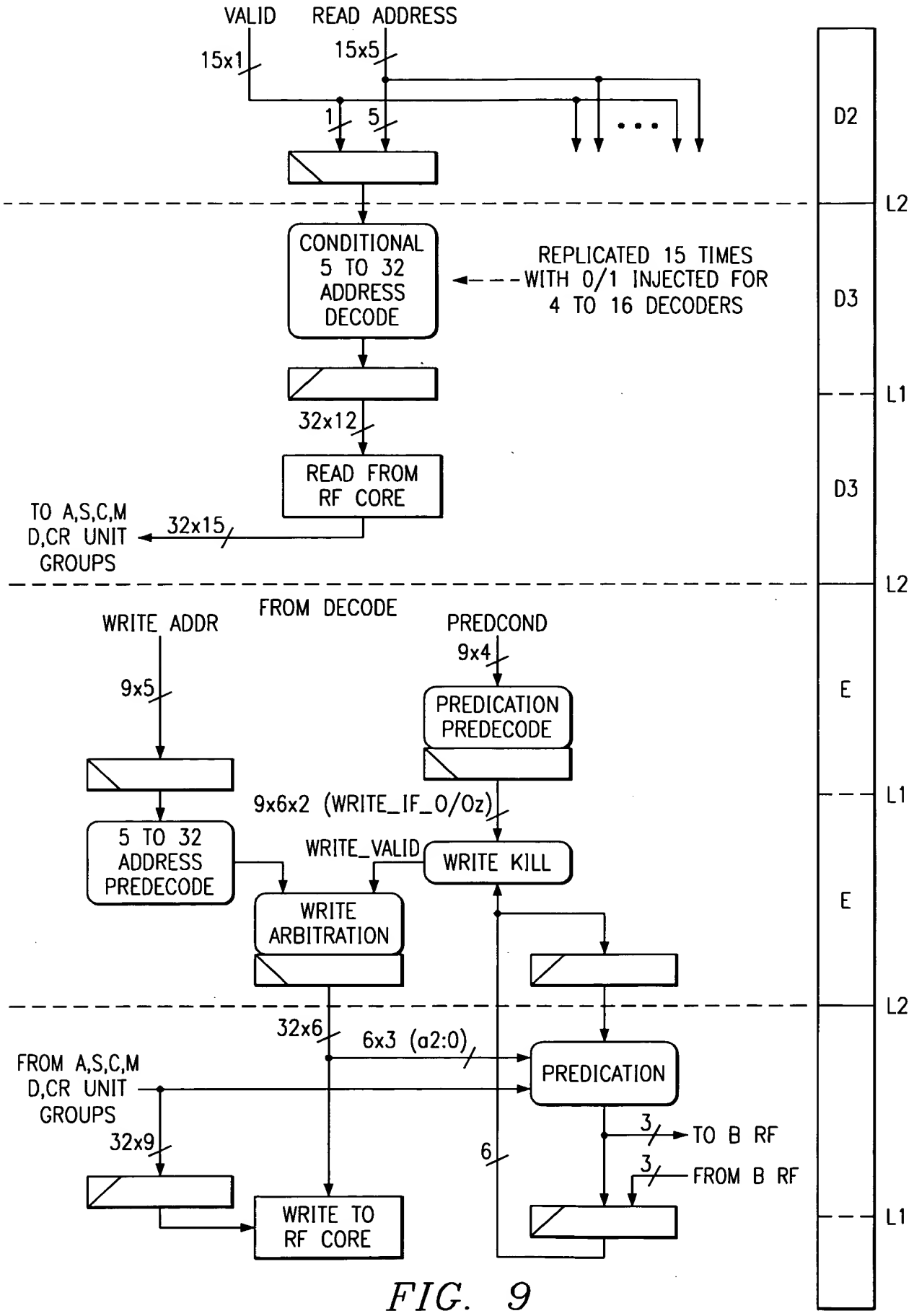

VALID
READ ADDRESS
15x1
15x5
1
5
D2
L2
CONDITIONAL 5 TO 32 ADDRESS DECODE
REPLICATED 15 TIMES WITH 0/1 INJECTED FOR 4 TO 16 DECODERS
D3
L1
32x12
READ FROM RF CORE
D3
TO A,S,C,M D,CR UNIT GROUPS
32x15
L2
FROM DECODE
WRITE ADDR
PREDCOND
9x4
9x5
PREDICATION PREDECODE
E
L1
9x6x2 (WRITE_IF_0/0z)
5 TO 32 ADDRESS PREDECODE
WRITE_VALID
WRITE KILL
WRITE ARBITRATION
E
L2
32x6
6x3 (a2:0)
PREDICATION
FROM A,S,C,M D,CR UNIT GROUPS
3
TO B RF
32x9
6
3
FROM B RF
WRITE TO RF CORE
L1

FIG. 9

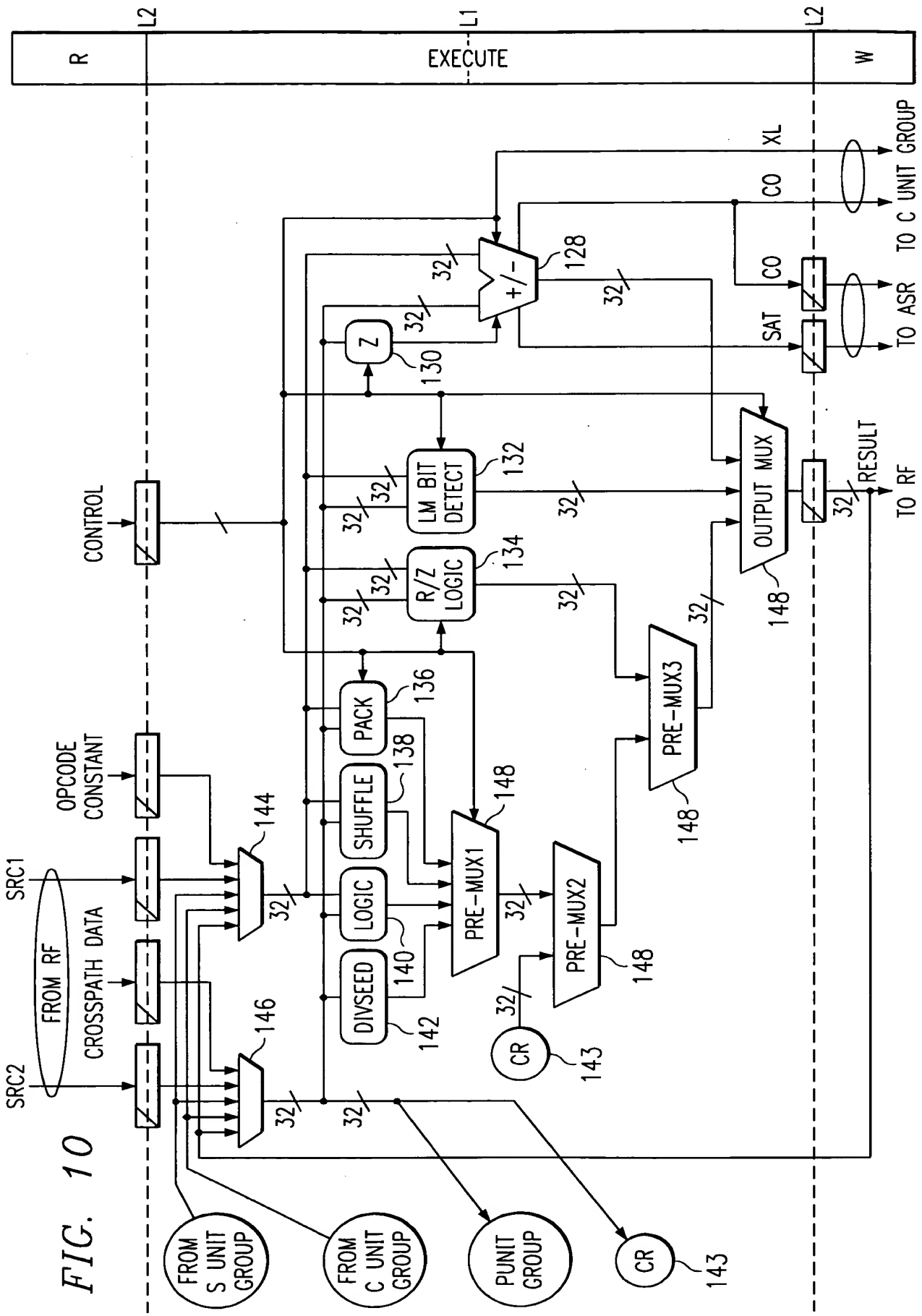

R
L2
EXECUTE
L1
W
L2
CONTROL
OPCODE CONSTANT
SRC1
FROM RF
CROSSPATH DATA
SRC2
144
146
32
DIVSEED
LOGIC
SHUFFLE
PACK
136
138
140
142
PRE-MUX1
148
PRE-MUX2
PRE-MUX3
CR
143
R/Z LOGIC
134
LM BIT DETECT
132
Z
130
+/-
128
OUTPUT MUX
XL
CO
SAT
TO C UNIT GROUP
TO ASR
RESULT
TO RF
FROM S UNIT GROUP
FROM C UNIT GROUP
PUNIT GROUP

FIG. 10

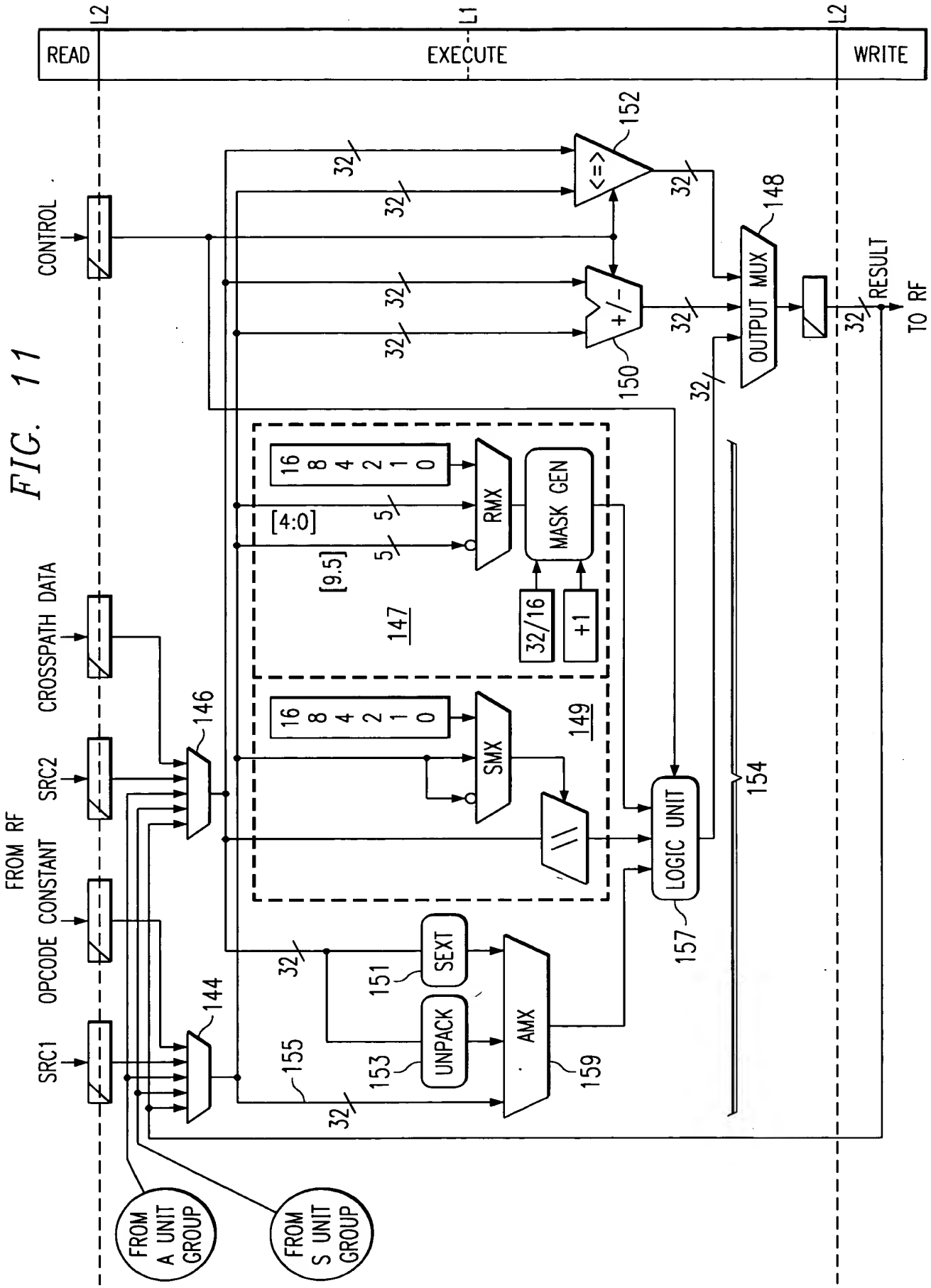

L2
L1
L2
READ
EXECUTE
WRITE
CONTROL
FROM RF
SRC1
OPCODE CONSTANT
SRC2
CROSSPATH DATA
FROM A UNIT GROUP
FROM S UNIT GROUP
144
146
147
149
150
151
152
153
154
155
157
159
148
32
<=>
+/-
OUTPUT MUX
RESULT
TO RF
16
8
4
2
1
0
[4:0]
[9.5]
5
RMX
MASK GEN
32/16
+1
SMX
SEXT
UNPACK
AMX
LOGIC UNIT

FIG. 11

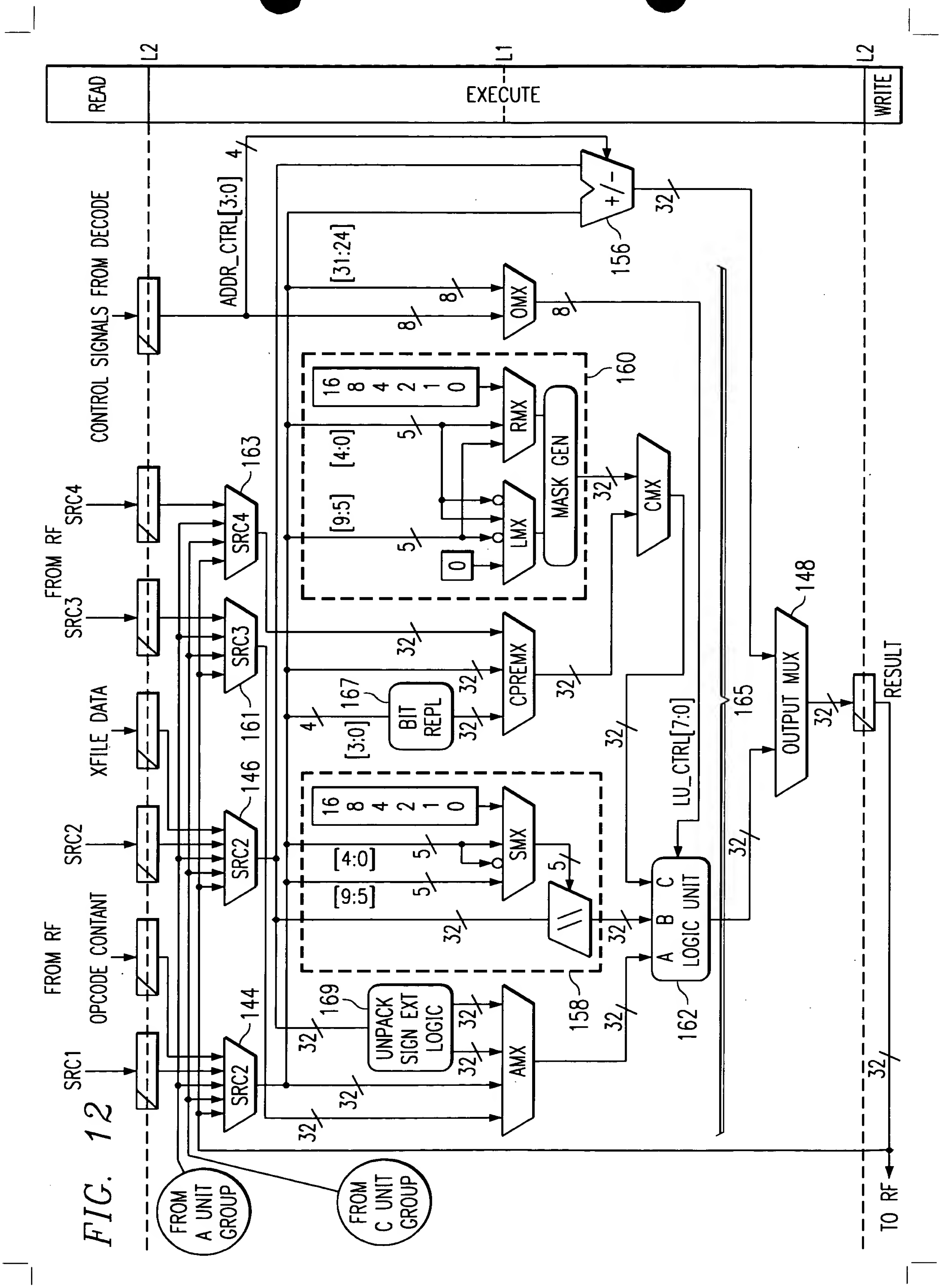

READ
EXECUTE
WRITE
L2
L1
L2
CONTROL SIGNALS FROM DECODE
FROM RF
SRC4
SRC3
XFILE DATA
SRC2
FROM RF
OPCODE CONTANT
SRC1
ADDR_CTRL[3:0]
4
+/-
156
32
[31:24]
8
OMX
8
8
160
16 8 4 2 1 0
RMX
MASK GEN
LMX
0
[4:0]
[9:5]
5
5
CMX
163
SRC4
SRC3
161
CPREMX
32
32
32
BIT REPL
167
4
[3:0]
148
OUTPUT MUX
RESULT
165
LU_CTRL[7:0]
146
SRC2
16 8 4 2 1 0
SMX
[4:0]
[9:5]
5
32
LOGIC UNIT
A B C
162
158
144
SRC2
169
UNPACK SIGN EXT LOGIC
AMX
32
FROM A UNIT GROUP
FROM C UNIT GROUP
TO RF

FIG. 12

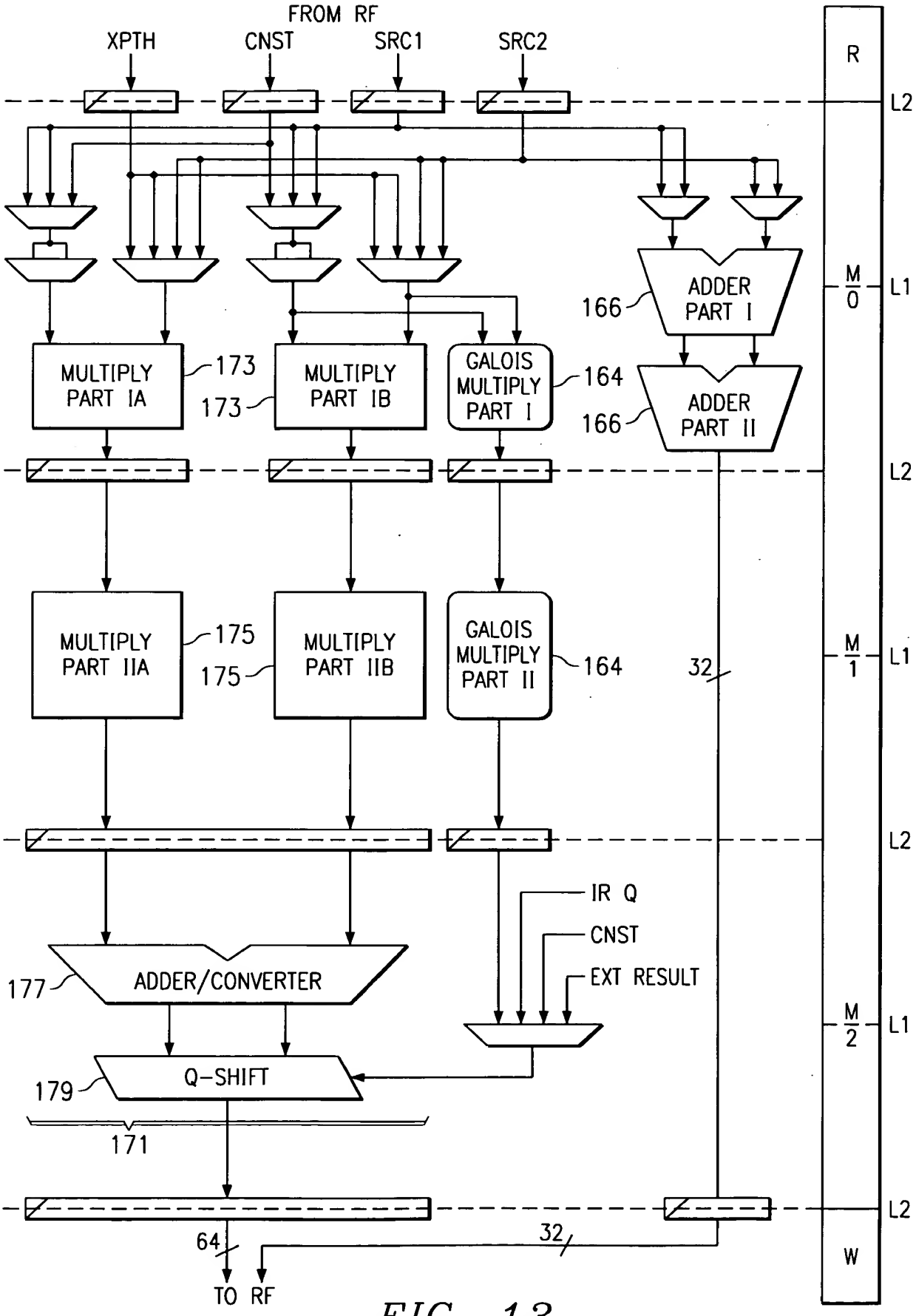

FROM RF
XPTH
CNST
SRC1
SRC2
R
L2
M/0
L1
ADDER PART I
166
MULTIPLY PART IA
173
MULTIPLY PART IB
173
GALOIS MULTIPLY PART I
164
ADDER PART II
166
L2
MULTIPLY PART IIA
175
MULTIPLY PART IIB
175
GALOIS MULTIPLY PART II
164
32
M/1
L1
L2
IR Q
CNST
EXT RESULT
ADDER/CONVERTER
177
M/2
L1
Q-SHIFT
179
171
L2
64
32
TO RF
W

FIG. 13

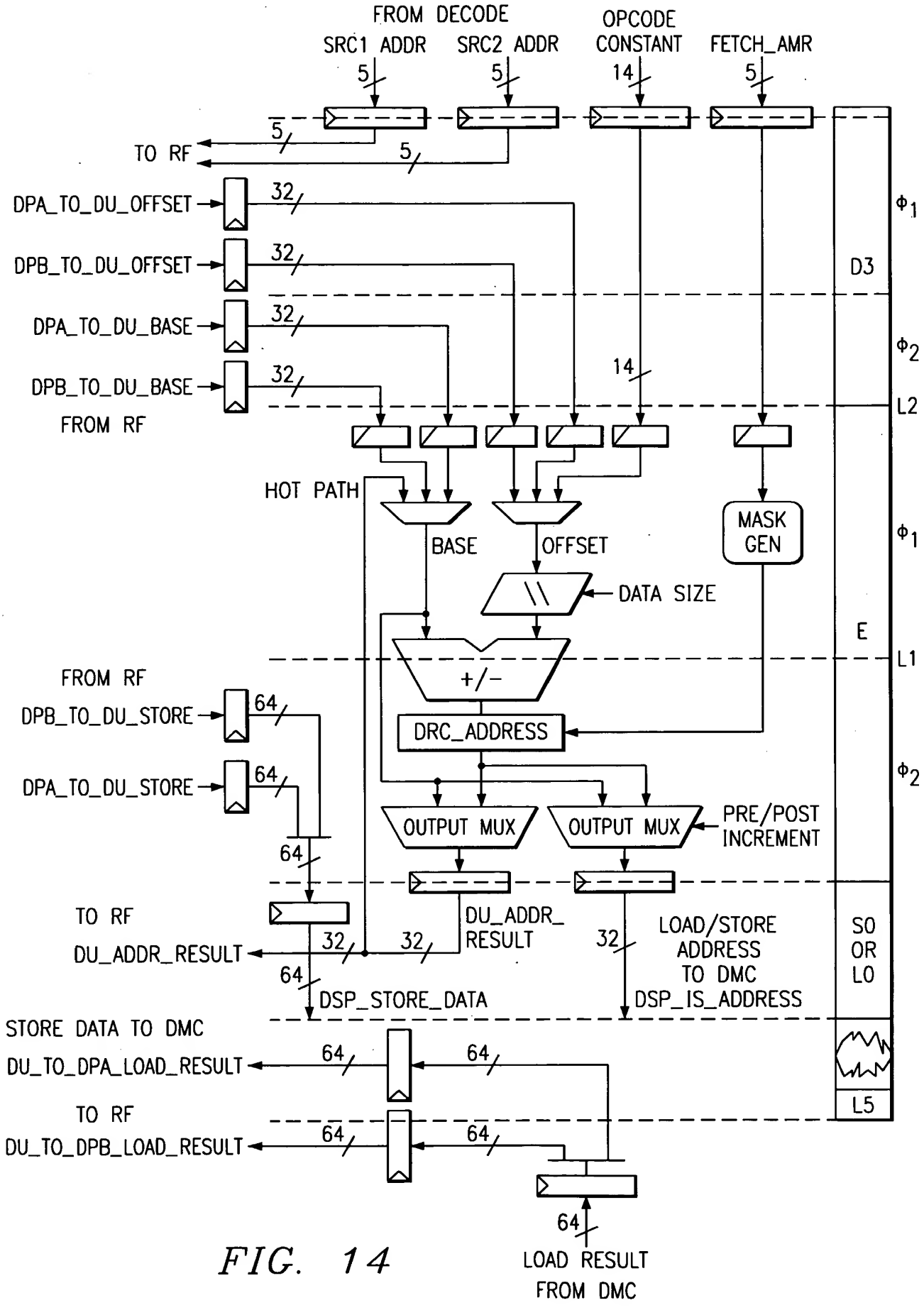

FROM DECODE
SRC1 ADDR
SRC2 ADDR
OPCODE CONSTANT
FETCH_AMR
5
5
14
5
TO RF
5
5
DPA_TO_DU_OFFSET
32
DPB_TO_DU_OFFSET
32
DPA_TO_DU_BASE
32
DPB_TO_DU_BASE
32
FROM RF
14
Φ1
D3
Φ2
L2
HOT PATH
BASE
OFFSET
MASK GEN
Φ1
DATA SIZE
E
L1
FROM RF
+/-
DPB_TO_DU_STORE
64
DRC_ADDRESS
DPA_TO_DU_STORE
64
Φ2
OUTPUT MUX
OUTPUT MUX
PRE/POST INCREMENT
64
TO RF
DU_ADDR_RESULT
32
32
DU_ADDR_RESULT
32
LOAD/STORE ADDRESS TO DMC
DSP_IS_ADDRESS
S0 OR L0
64
DSP_STORE_DATA
STORE DATA TO DMC
DU_TO_DPA_LOAD_RESULT
64
64
TO RF
DU_TO_DPB_LOAD_RESULT
64
64
L5
64
LOAD RESULT FROM DMC

FIG. 14

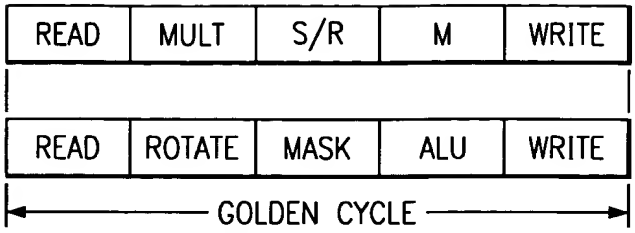


FIG. 16a
(PRIOR ART)

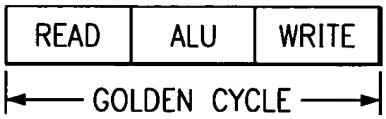


FIG. 16b
(PRIOR ART)

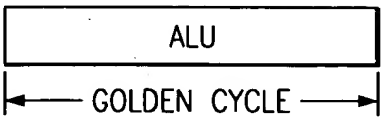


FIG. 16c

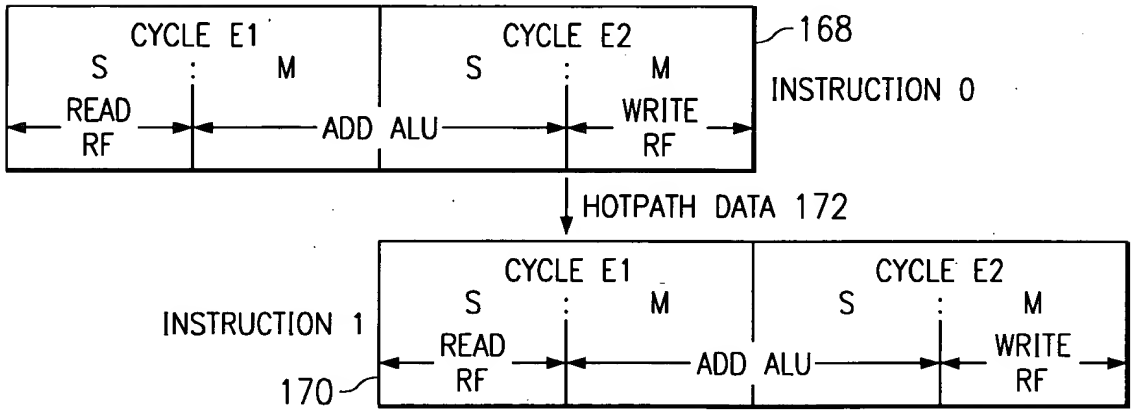


FIG. 17

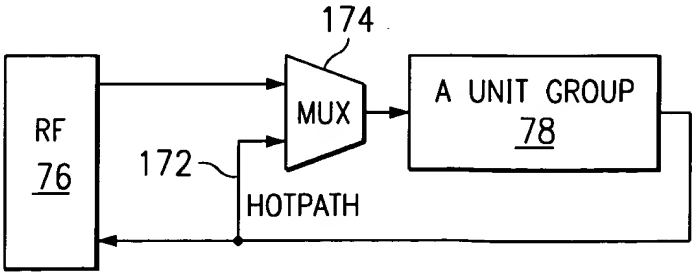
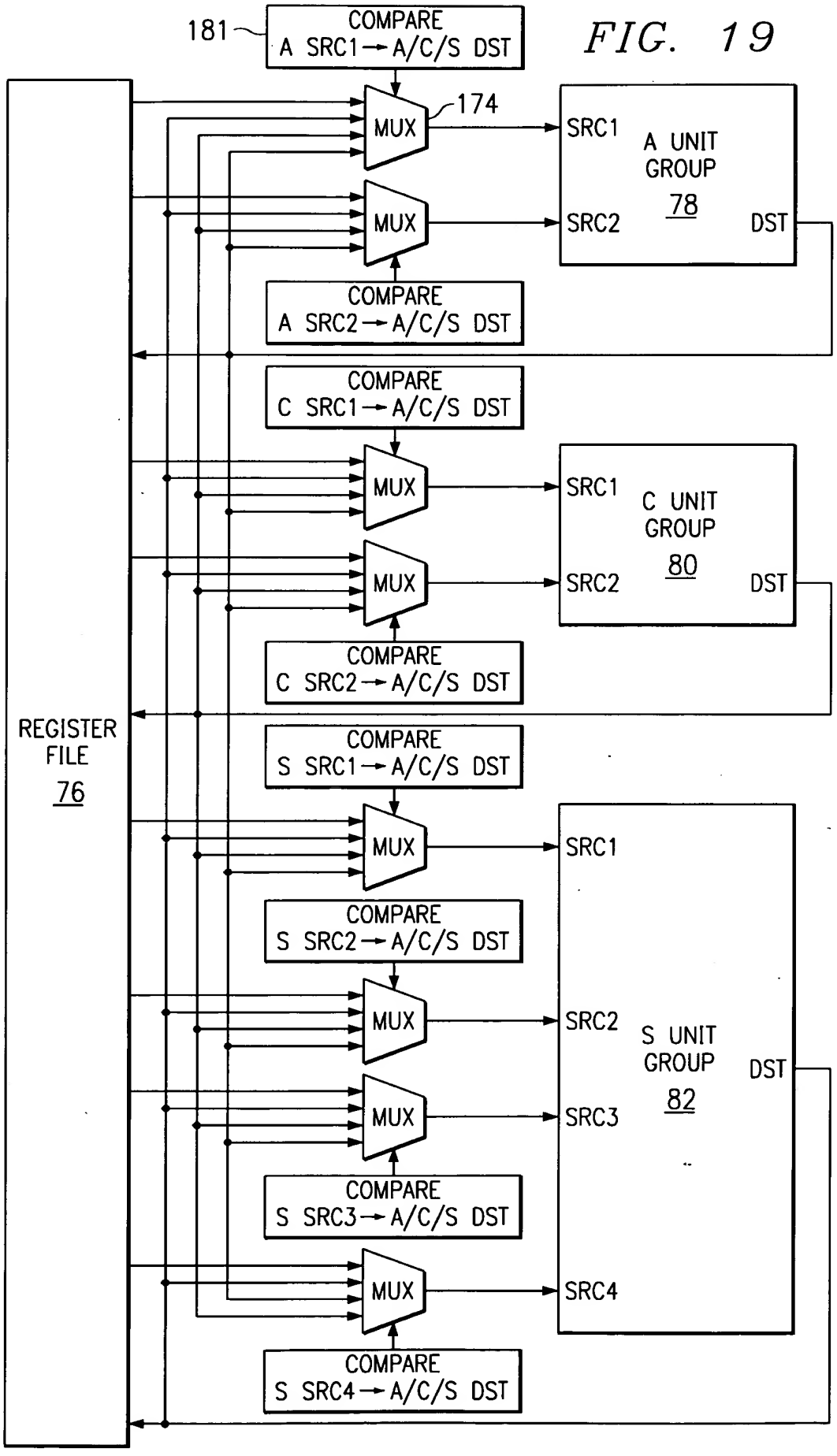


FIG. 18

FIG. 19


181
COMPARE
A SRC1→A/C/S DST
174
MUX
MUX
SRC1
SRC2
A UNIT
GROUP
78
DST
COMPARE
A SRC2→A/C/S DST
COMPARE
C SRC1→A/C/S DST
MUX
MUX
SRC1
SRC2
C UNIT
GROUP
80
DST
COMPARE
C SRC2→A/C/S DST
REGISTER
FILE
76
COMPARE
S SRC1→A/C/S DST
MUX
SRC1
COMPARE
S SRC2→A/C/S DST
MUX
SRC2
MUX
SRC3
S UNIT
GROUP
82
DST
COMPARE
S SRC3→A/C/S DST
MUX
SRC4
COMPARE
S SRC4→A/C/S DST

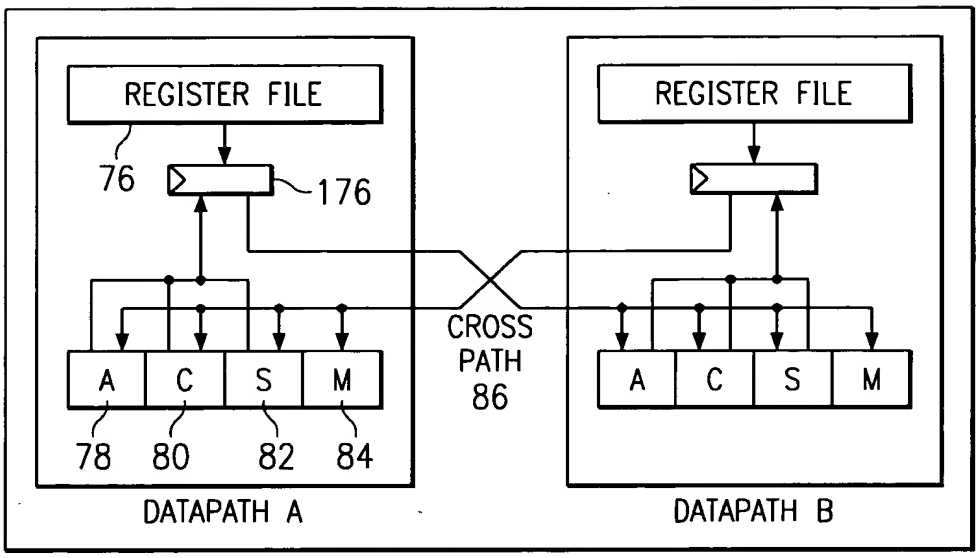


FIG. 20

		SOURCE(S) DATA PATH					
		A,C,S	M (ADD)	M (MUL)	D (LOAD)	D (STORE)	D (ADDR)
SINK(S) IN SAME DATAPATH	A,C,S	0	1	3	7	–	1
	M	1	1	3	7	–	1
	D (LOAD)	–	–	–	–	0	–
	D (STORE)	0	0	2	6	–	0
	D (ADDR:BASE)	1	–	3	7	–	0
	D (ADDR:OFFSET)	1	–	3	7	–	1
	P	0	1	3	7	–	1
	PREDICATION	0	–	2	6	–	0
SINK(S) IN OPPOSITE DATAPATH	A,C,S	1	2	4	8	–	2
	M	1	2	4	8	–	2
	PREDICATION	0	–	2	6	–	0

FIG. 21

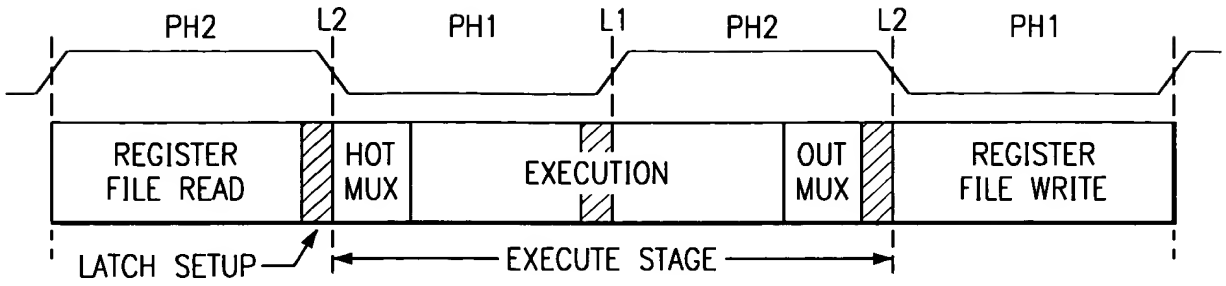


FIG. 22a

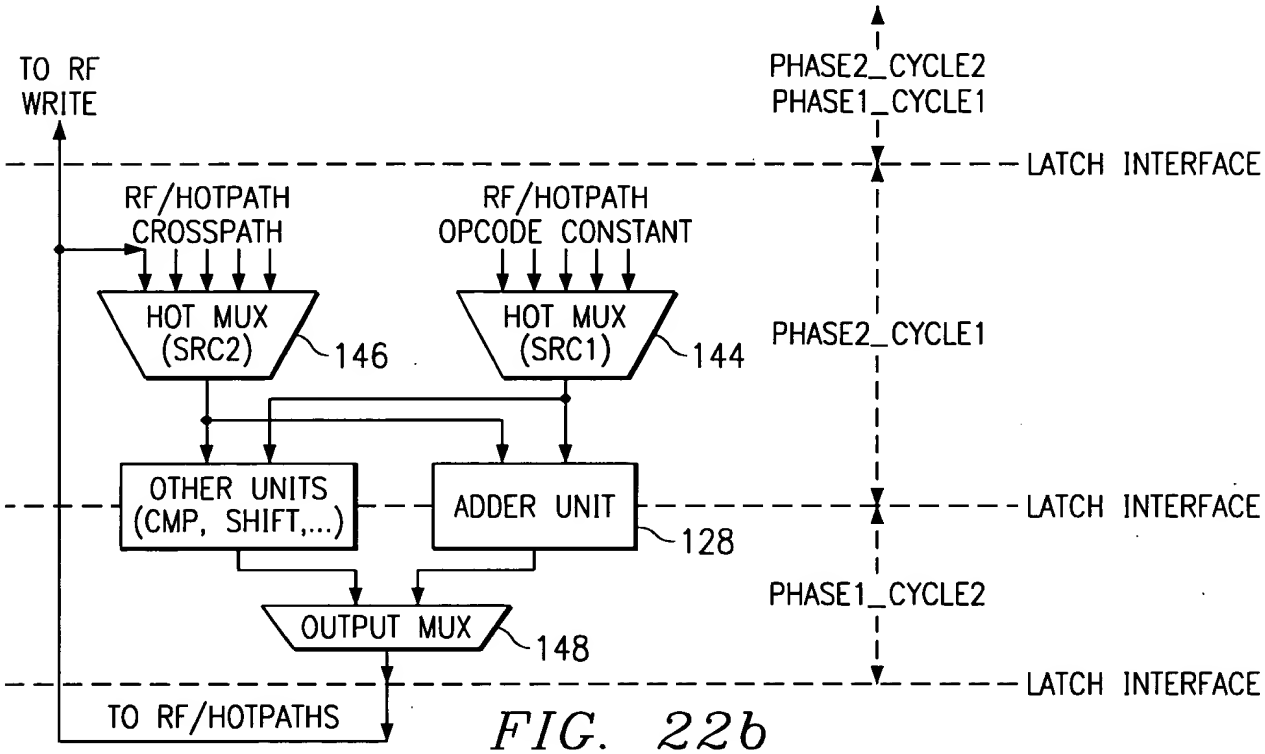


FIG. 22b

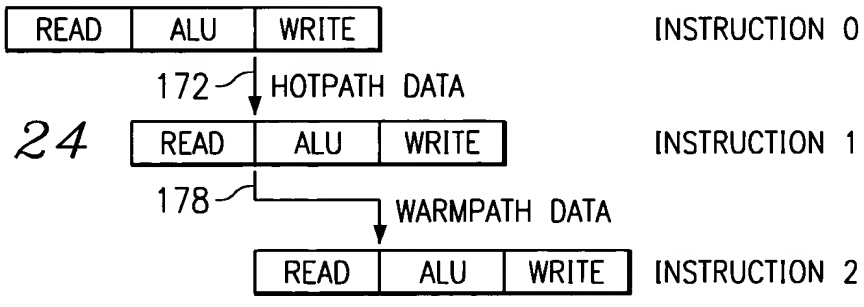


FIG. 24

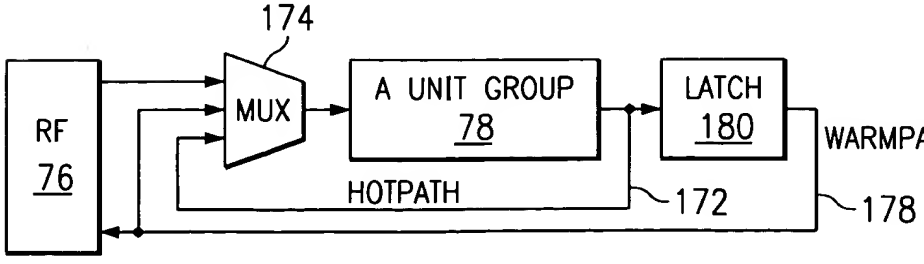
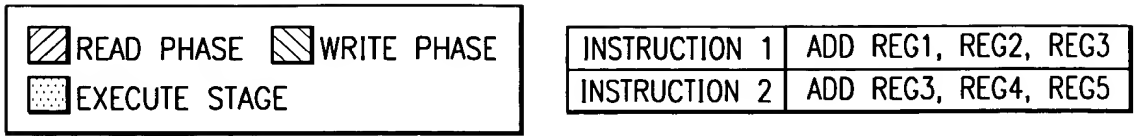


FIG. 25



HOTPATH AVAILABLE

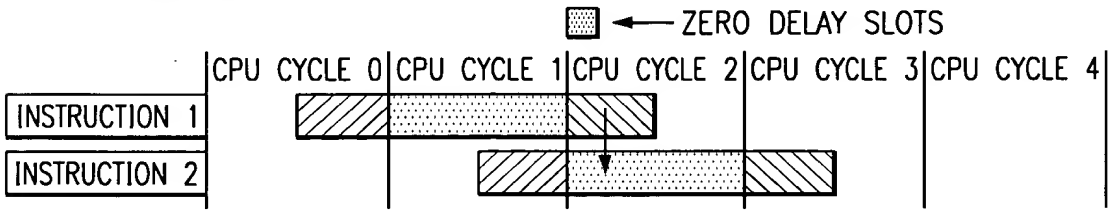


FIG. 23a

HOTPATH UNAVAILABLE

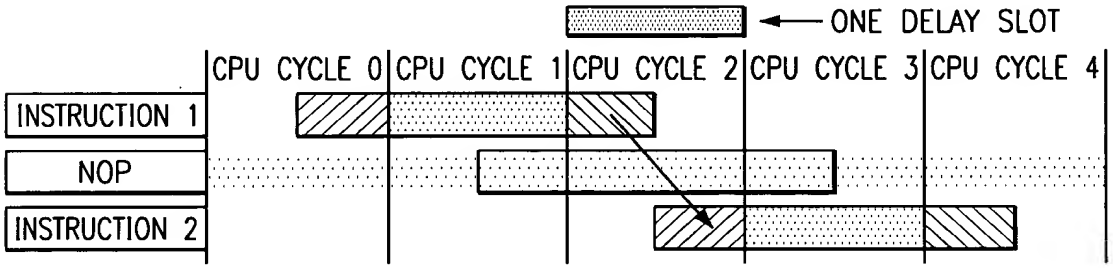


FIG. 23b

CROSSFILE ACCESS WITH HOTPATH AVAILABLE

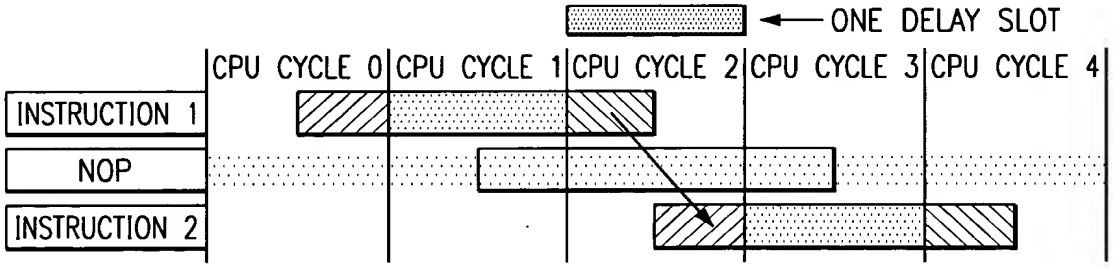


FIG. 23c

CROSSFILE ACCESS WITH HOTPATH UNAVAILABLE

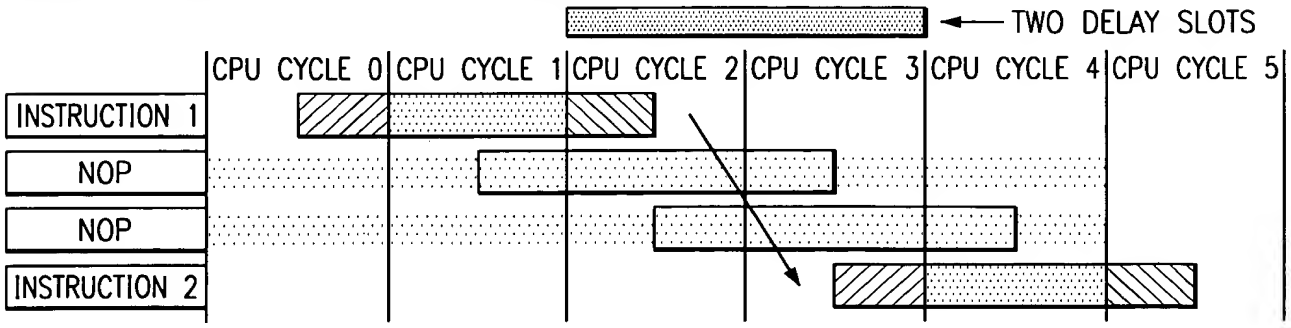


FIG. 23d